

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

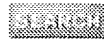
As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.



US Patent & Trademark Office

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

xml transformation and data loss and transformation cost

Found 57,498 of 134,837

Sort results by

relevance

[Save results to a Binder](#)[Try an Advanced Search](#)

Display results

expanded form

[Search Tips](#)[Try this search in The ACM Guide](#)
☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Web Information Integration: Automating the transformation of XML documents](#)

Hong Su, Harumi Kuno, Elke A. Rundensteiner

November 2001 **Proceeding of the third international workshop on Web information and data management**

Full text available: pdf(522.82 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The advent of web services that use XML-based message exchanges has spurred many efforts that address issues related to inter-enterprise service electronic commerce interactions. Currently emerging standards and technologies enable enterprises to describe and advertise their own Web Services and to discover and determine how to interact with services fronted by other businesses. However, these technologies do not address the problem of how to reconcile structural differences between similar type ...

2 [Special section on advanced XML data processing: Preservation of digital data with self-validating, self-instantiating knowledge-based archives](#)

Bertram Ludäscher, Richard Marciano, Reagan Moore

September 2001 **ACM SIGMOD Record**, Volume 30 Issue 3

Full text available: pdf(881.20 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Digital archives are dedicated to the long-term preservation of electronic information and have the mandate to enable sustained access despite rapid technology changes. Persistent archives are confronted with heterogeneous data formats, helper applications, and platforms being used over the lifetime of the archive. This is not unlike the interoperability challenges, for which mediators are devised. To prevent technological obsolescence over time and across platforms, a migration approach for per ...

3 [Session 4: Web service applications: Authenticating distributed data using Web services and XML signatures](#)

Daniel J. Polivy, Roberto Tamassia

November 2002 **Proceedings of the 2002 ACM workshop on XML security**

Full text available: pdf(164.09 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

As the need for digital data becomes more ubiquitous, so does the need to provide efficient mechanisms for distributing and verifying the authenticity of that data. We present an architecture for authenticating responses to queries from untrusted mirrors of authenticated dictionaries using Web Services and XML Signatures. We also describe an implementation of

our scheme for the Secure Transaction Management System.

Keywords: Web services, XML, authentication, digital signatures

4 Applications: Price modeling in standards for electronic product catalogs based on XML 

Oliver Kelkar, Joerg Leukel, Volker Schmitz

May 2002 **Proceedings of the eleventh international conference on World Wide Web**

Full text available:  pdf(239.41 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


The fast spreading of electronic business-to-business procurement systems has led to the development of new standards for the exchange of electronic product catalogs (e-catalogs). E-catalogs contain various information about products, essential is price information. Prices are used for buying decisions and following order transactions. While simple price models are often sufficient for the description of indirect goods (e.g. office supplies), other goods and lines of business make higher demands ...

Keywords: B2B, XML, e-business, e-catalog, e-procurement, pricing

5 Research sessions: XML II: Approximate XML joins 

Sudipto Guha, H. V. Jagadish, Nick Koudas, Divesh Srivastava, Ting Yu

June 2002 **Proceedings of the 2002 ACM SIGMOD international conference on Management of data**

Full text available:  pdf(1.25 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

XML is widely recognized as the data interchange standard for tomorrow, because of its ability to represent data from a wide variety sources. Hence, XML is likely to be the format through which data from multiple sources is integrated. In this paper we study the problem of integrating XML data sources through correlations realized as join operations. A challenging aspect of this operation is the XML document structure. Two documents might convey approximately or exactly the same information but m ...

6 XML query processing II: On relational support for XML publishing: beyond sorting and tagging 

Surajit Chaudhuri, Raghav Kaushik, Jeffrey F. Naughton

June 2003 **Proceedings of the 2003 ACM SIGMOD international conference on Management of data**


Full text available:  pdf(237.72 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In this paper, we study whether the need for efficient XML publishing brings any new requirements for relational query engines, or if sorting query results in the relational engine and tagging them in middleware is sufficient. We observe that the mismatch between the XML data model and the relational model requires relational engines to be enhanced for efficiency. Specifically, they need to support relation valued variables. We discuss how such support can be provided through the addition of an ...

7 Standardization in IT: Inter-organizational document exchange: facing the conversion problem with XML 

Luis Martín Díaz, Erik Wüstner, Peter Buxmann

March 2002 **Proceedings of the 2002 ACM symposium on Applied computing**

Full text available:  pdf(470.62 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Information exchange processes are often characterized by the need of translating from one data format into another in order to achieve compatibility between information systems. A conversion problem often arises when exchanging files between applications of different software vendors or when incorporating legacy business data into new standard software. In this paper we want to survey the conversion problem in the field of multi-organizational networks, since participants often use different da ...

Keywords: Java, XML, conversion problem, information systems, inter-organizational document exchange, standardization, supply chain management

- 8 XML indexing and compression: XPRESS: a queriable compression for XML data
Jun-Ki Min, Myung-Jae Park, Chin-Wan Chung
June 2003 **Proceedings of the 2003 ACM SIGMOD international conference on on Management of data**

Full text available:  [pdf\(277.17 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Like HTML, many XML documents are resident on native file systems. Since XML data is irregular and verbose, the disk space and the network bandwidth are wasted. To overcome the verbosity problem, the research on compressors for XML data has been conducted. However, some XML compressors do not support querying compressed data, while other XML compressors which support querying compressed data blindly encode tags and data values using predefined encoding methods. Thus, the query performance on com ...


- 9 Design and development of data-intensive web sites: The Araneus approach
Paolo Merialdo, Paolo Atzeni, Giansalvatore Mecca
February 2003 **ACM Transactions on Internet Technology (TOIT)**, Volume 3 Issue 1

Full text available:  [pdf\(2.18 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Data-intensive Web sites are large sites based on a back-end database, with a fairly complex hypertext structure. The paper develops two main contributions: (a) a specific design methodology for data-intensive Web sites, composed of a set of steps and design transformations that lead from a conceptual specification of the domain of interest to the actual implementation of the site; (b) a tool called Homer, conceived to support the site design and implementation process, by allowing the ...

Keywords: Databases, Internet, WWW, World Wide Web, development

- 10 Answering queries using views: A survey
Alon Y. Halevy
December 2001 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 10 Issue 4

Full text available:  [pdf\(308.74 KB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

The problem of answering queries using views is to find efficient methods of answering a query using a set of previously defined materialized views over the database, rather than accessing the database relations. The problem has recently received significant attention because of its relevance to a wide variety of data management problems. In query optimization, finding a rewriting of a query using a set of materialized views can yield a more efficient query execution plan. To support the separat ...

Keywords: Data integration, Date warehousing, Materialized views, Query optimization, Survey, Web-site management

Semantic extensions of XML for advanced applications


Kazumasa Yokota, Takeo Kunishima, Bojiang Liu

January 2001 **Australian Computer Science Communications**, Volume 23 Issue 6Full text available:  pdf(830.05 KB)Additional Information: [full citation](#), [abstract](#), [references](#) [Publisher Site](#)


XML is a *de facto* standard for structuring documents and their exchange in electronic forms. However it has many restrictions to represent various information from viewpoints of data-centered applications. Set and tuple constructors are useful for defining data structure, and identities and user views are indispensable for sharing common data, while XML, even with XML Schema[16], does not support such features. In this paper, we propose an approach to semantic extensions of XML by introduc ...

12 Type inference for queries on semistructured data

Tova Milo, Dan Suciu

May 1999 **Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems**Full text available:  pdf(1.37 MB)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**13 Formally optimal boxing**

Fritz Henglein, Jesper Jørgensen

February 1994 **Proceedings of the 21st ACM SIGPLAN-SIGACT symposium on Principles of programming languages**Full text available:  pdf(1.53 MB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

An important implementation decision in polymorphically typed functional programming language is whether to represent data in boxed or unboxed form and when to transform them from one representation to the other. Using a language with explicit representation types and boxing/unboxing operations we axiomatize equationally the set of all explicitly boxed versions, called completions, of a given source program. In a two-stage process we give some of the equations a rewriting i ...

Keywords: polymorphism, representation analysis, type inference

14 XML query processing I: Dynamic XML documents with distribution and replication

Serge Abiteboul, Angela Bonifati, Grégory Cobéna, Ioana Manolescu, Tova Milo

June 2003 **Proceedings of the 2003 ACM SIGMOD international conference on Management of data**Full text available:  pdf(209.06 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The advent of XML as a universal exchange format, and of Web services as a basis for distributed computing, has fostered the apparition of a new class of documents: *dynamic XML documents*. These are XML documents where some data is given explicitly while other parts are given only intensionally by means of embedded calls to web services that can be called to generate the required information. By the sole presence of Web services, dynamic documents already include inherently some form of di ...


15 Reconciling schemas of disparate data sources: a machine-learning approach

AnHai Doan, Pedro Domingos, Alon Y. Halevy

May 2001 **ACM SIGMOD Record , Proceedings of the 2001 ACM SIGMOD international conference on Management of data**, Volume 30 Issue 2

Full text available:

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)


 pdf(366.64 KB)[terms](#)

A data-integration system provides access to a multitude of data sources through a single mediated schema. A key bottleneck in building such systems has been the laborious manual construction of semantic mappings between the source schemas and the mediated schema. We describe LSD, a system that employs and extends current machine-learning techniques to semi-automatically find such mappings. LSD first asks the user to provide the semantic mappings for a small set of data sources, then uses the ...

16 DDD papers: Software factories: assembling applications with patterns, models, frameworks and tools

Jack Greenfield, Keith Short

October 2003 **Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**

Full text available:  pdf(707.51 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


The confluence of component based development, model driven development and software product lines forms an approach to application development based on the concept of software factories. This approach promises greater gains in productivity and predictability than those produced by incremental improvements to the current paradigm of object orientation, which have not kept pace with innovation in platform technology. Software factories promise to make application assembly more cost effective thro ...

Keywords: design patterns, domain-specific languages, model-driven development, software factories, software product lines

17 Distributed query evaluation on semistructured data

Dan Suciu

March 2002 **ACM Transactions on Database Systems (TODS)**, Volume 27 Issue 1

Full text available:  pdf(689.88 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Semistructured data is modeled as a rooted, labeled graph. The simplest kinds of queries on such data are those which traverse paths described by regular path expressions. More complex queries combine several regular path expressions, with complex data restructuring, and with sub-queries. This article addresses the problem of efficient query evaluation on distributed, semistructured databases. In our setting, the nodes of the database are distributed over a fixed number of sites, and the ...

Keywords: Distributed evaluation, nested queries, parallel complexity, regular expressions, semistructured data

18 Software: An open software architecture for virtual reality interaction

Gerhard Reitmayr, Dieter Schmalstieg

November 2001 **Proceedings of the ACM symposium on Virtual reality software and technology**

Full text available:  pdf(348.46 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


This article describes OpenTracker, an open software architecture that provides a framework for the different tasks involved in tracking input devices and processing multi-modal input data in virtual environments and augmented reality application. The OpenTracker framework eases the development and maintenance of hardware setups in a more flexible manner than what is typically offered by virtual reality development packages. This goal is achieved by using an object-oriented design based on XML, ...

Keywords: XML, mobile augmented reality, tracking, virtual reality

19 Location awareness and moving objects: A data repository for fine-grained adaptation in heterogeneous environments

Calicrates Policroniades, Rajiv Chakravorty, Pablo Vidales

September 2003 **Proceedings of the 3rd ACM international workshop on Data engineering for wireless and mobile access**

Full text available:  pdf(145.68 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


In this paper, we present DATOM -- an extended structured data storage system for mobile data management. We propose cooperation of networks and applications in order to adapt to the limitations imposed by heterogeneous environments. We show how adaptation can be achieved to address the high variability in link-layer characteristics as typically seen in hybrid wireless networks. We propose a novel data model for DATOM, that departs from the traditional view of files as monolithic objects, and bre ...

Keywords: data adaptation, mobile computing, overlay networks

20 ObjectGlobe: Ubiquitous query processing on the Internet

R. Braumandl, M. Keidl, A. Kemper, D. Kossmann, A. Kreutz, S. Seltzsam, K. Stocker

August 2001 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 10 Issue 1

Full text available:  pdf(251.44 KB) Additional Information: [full citation](#), [abstract](#), [index terms](#)

We present the design of ObjectGlobe, a distributed and open query processor for Internet data sources. Today, data is published on the Internet via Web servers which have, if at all, very localized query processing capabilities. The goal of the ObjectGlobe project is to establish an open marketplace in which *data* and *query processing capabilities* can be distributed and used by any kind of Internet application. Furthermore, ObjectGlobe integrates *cycle providers* (i.e., machi ...

Keywords: Cycle-, function- and data provider, Distributed query processing, Open systems, Privacy, Quality of service, Query optimization, Security

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

First Hit

Generate Collection

Print

L4: Entry 7 of 30

File: PGPB

Aug 7, 2003

DOCUMENT-IDENTIFIER: US 20030149934 A1

TITLE: Computer program connecting the structure of a xml document to its underlying meaning

Summary of Invention Paragraph:

[0018] A similar problem of interoperability arose in the 1980s with the emergence of relational databases. In spite of the existence of an underlying technology to solve it (Relational Views), it has in practice not been solved in twenty years. The result has been an information Babel within every major company, which has multiplied their information management and IT development costs by a large factor.

Detail Description Paragraph:

[0070] So whenever a new XML language comes along--as will frequently happen--all you need do is find (or if need be, write down) the MDL definition of that language. Then all your systems and users, using that MDL, will be immediately adapted to the new language, without any further effort. As XML usage grows and languages proliferate, the cost-savings from this easy adaptation will be huge.

Detail Description Paragraph:

[0181] Developers are interested in getting the meaning out of an XML document (or putting it in). To do this via DOM, they need to understand the XML document structure, and how it conveys meanings, quite precisely. For large and complex XML languages, this is costly and error-prone.

Detail Description Paragraph:

[0213] As XML languages continue to proliferate, we believe that the benefits of this meaning-level style of application development--in quality, development costs and maintenance costs--will be overwhelming. There is no reason not to start doing it now.

Detail Description Paragraph:

[0216] This has the same drawbacks for query users as the structure-level APIs have for developers. Users need to understand the structure of XML languages--which for large languages may be costly and error-prone--and queries are not transportable across XML languages.

Detail Description Paragraph:

[0251] Compared with the meaning-driven approach, writing and debugging of XSLT is much more expensive and error-prone. Even to write one XSLT translation is, we believe, more costly than to write down the MDL for the two languages involved. The XSLT is generally a much larger and more complex document than the two MDL files; and in many cases you will already have the MDL files available.

Detail Description Paragraph:

[0252] However, it is when there are several different languages that the advantages of the MDL approach become overwhelming. With N different languages, you may require as many as $N*(N-1)$ distinct translations between them. Using MDL, the cost of creating all these translations grows only as N (this is the cost of writing all the MDL files). This can rapidly amount to a huge cost difference--

especially as each different language may go through a series of versions.

Detail Description Paragraph:

[0254] XML mapping tools such as Biztalk Mapper display two tree diagrams side by side, showing the element nesting structures of two XML languages. The user can then drag-and-drop from one tree to the other, to define 'mappings' between the two languages, and these mappings are used to generate an XSLT translation between them. However, this simple node-to-node mapping technique does not capture all the ways in which the two XML languages may represent associations; therefore it is not capable of translating association information correctly. For instance, if one language represents an association by shared values, while the other represents the same association by element nesting, tools like BizTalk Mapper cannot do faithful translations in both directions. Since association information is a vital part of XML content, and XML languages represent associations in a wide variety of ways, this means that XML-to-XML mapping tools will fail for many important translation tasks. Furthermore, since these tools require mappings to be defined afresh for each pair of languages, the cost of creating all possible translations between N languages grows as $N*(N-1)$, rather than N.

Detail Description Paragraph:

[0336] XMuLator XML Transformation Tool

Detail Description Paragraph:

[0349] A similar problem of interoperability arose in the 1980s with the emergence of relational databases. In spite of the existence of an underlying technology to solve it (Relational Views), it has in practice not been solved in twenty years. The result has been an information Babel within every major company, which has multiplied their information management and IT development costs by a large factor.

Detail Description Paragraph:

[0390] Unbundling and Normalisation: Many computer file structures and data structures (for instance, many classes in object-oriented programming) bundle together information about several different types of thing together in the same object or file record. XML messages typically bundle a lot inside one element. In contrast, the business information model is maximally unbundled (or in relational database terminology, normalised) to make it absolutely clear what information pertains to what kind of entity. It should be so, to be able to represent the business realistically and flexibly, and it can be so, because it is a tool for analysis, and does not have to be 'optimised' for performance. Most of the bundled computing structures have been bundled partly for reasons of performance, partly for implementation simplicity in a specific application. This bundling typically has unforeseen costs when the application is broadened or altered.

Detail Description Paragraph:

[0429] The business information model is a taxonomy of entity classes, with attributes and relations. You may be concerned that you need to 'get this model right'--in particular, to get the taxonomy structure right--before you can start using it to generate XML transformations. For two reasons, this is not the case.

Detail Description Paragraph:

[0431] As far as XML transformation is concerned, these multiple definitions do not matter. As long as two different XML languages represent the same class, attribute, or relation, that information can be translated between them--wherever it is defined on the taxonomy.

Detail Description Paragraph:

[0432] For the same reason, the lack of multiple inheritance in the XMuLator business model does not stop you generating good XML transformations--it just means you may need to define an attribute or relation in several places, where multiple

inheritance would have allowed you to define it just once.

Detail Description Paragraph:

[0522] Sometimes it is only possible to provide a template to convert in one direction, because information is lost in conversion and cannot be recovered. For instance, a representation of a full name which uses a middle initial cannot be converted back to recover the middle name. XMuLator will then be able to convert in one direction only.

Detail Description Paragraph:

[0587] Business information consists of classes (of entities), attributes and relations. Each of these parts of the business information model can be represented in an XML language, and can be so represented in a variety of ways. It is this variety of the ways in which XML can represent business information which makes the XML transformation problem difficult. Two different languages may represent the same business information in different ways, and it is necessary to transform between them while preserving the underlying business information.

Detail Description Paragraph:

[0941] If you have enough XML based languages, you can make long round trips through five or more languages. However, these long round trips are generally not a very sensitive test of the translations, because so much information gets lost of the way round. It seems more effective to test a variety of round trips through two, three and four languages at a time.

Detail Description Paragraph:

[0944] Building a business process model is not directly relevant to XML transformation, which depends only on the declared meanings of entity classes attributes and relations, and on the mappings of these to XML structure. However, the process model is often a very important underpinning of the meanings of things in the information model, since it defines how these things are used. It is therefore worth taking time to build a business process model and relate it to the business information model.

Detail Description Paragraph:

[0950] The set of information about each process which XMuLator can capture is quite open-ended; different attributes of a process can be built into the model at will. Typical information held about each process may include the role responsible for carrying out the process, the number of times the process is carried out, its typical costs and elapsed time.

Detail Description Paragraph:

[0962] In this map database, the only detail information held for a process (besides its description) is the Responsible Role. Depending on how a map database is set up, other detail information (such as the frequency or cost of a process) can be entered and shown here. Section 9 describes how to set up a map database to hold such extra information.

First Hit Fwd Refs☐ **Generate Collection** **Print**

L2: Entry 7 of 10

File: USPT

Nov 12, 2002

DOCUMENT-IDENTIFIER: US 6480865 B1

TITLE: Facility for adding dynamism to an extensible markup language

Brief Summary Text (23):

However, XML itself does not provide any mechanisms to perform such transformations, since XML documents are by themselves static in nature. There is no way to embed logic, or specify filtering, or specify transformation structures that are based upon the context in which the document is processed. Moreover, the concepts of applets and servlets as used in HTML are limited in that the APPLET and SERVLET tags occur at the leaf level of the HTML page, which means that there is no way to embed anything within an APPLET or SERVLET tag. What is needed, then, is a mechanism that seamlessly mixes programming language constructs with the constructs of XML so that one can reference and enable the other. Moreover, such mechanisms should exploit the fact that XML does not attach any specific semantics to its tags.

Current US Original Classification (1):715/523Current US Cross Reference Classification (1):715/512Current US Cross Reference Classification (2):715/513

CLAIMS:

1. A computer-implemented method for transforming an eXtensible Markup Language (XML) document, comprising: (a) identifying, in a computer, one or more annotations within the XML document, wherein: (i) the XML document is specified in a first namespace; (ii) the annotations are specified in a second namespace within the XML document that is different from the first namespace; (iii) a dynamic processor in the computer is configured to identify the annotations as belonging to the second namespace; (b) the dynamic processor invoking, in the computer, one or more functions associated with the identified annotations belonging to the second namespace, wherein: (i) the functions perform one or more transformations on the XML document to produce a transformed XML document; (ii) the transformed XML document replaces one or more portions of the XML document; and (iii) the transformed XML document can be processed by an XML processor.

19. An apparatus for transforming an eXtensible Markup Language (XML) document, comprising: (a) a computer; and (b) dynamic processor means, performed by the computer, for identifying one or more annotations within the XML document, wherein: (i) the XML document is specified in a first namespace; (ii) the annotations are specified in a second namespace within the XML document that is different from the first namespace; (iii) the dynamic processor means is configured to identify the annotations as belonging to the second namespace; and (iv) the dynamic processor is configured to invoke one or more functions associated with the identified annotations belonging to the second namespace, wherein: (1) the functions perform one or more transformations on the XML document to produce a transformed XML

document; (2) the transformed document replaces one or more portions of the XML document; and (3) the transformed document can be processed by an XML processor.

37. An article of manufacture embodying logic for performing a method for transforming an eXtensible Markup Language (XML) document, the method comprising: (a) identifying, in a computer, one or more annotations within the XML document, wherein: (i) the XML document is specified in a first namespace; (ii) the annotations are specified in a second namespace within the XML document that is different from the first namespace; (iii) a dynamic processor in the computer is configured to identify the annotations as belonging to the second namespace; and (b) the dynamic processor invoking, in the computer, one or more functions associated with the identified annotations belonging to the second namespace, wherein: (i) the functions perform one or more transformations on the XML document to produce a transformed XML document; (ii) the transformed XML document replaces one or more portions of the XML document; and (iii) the transformed XML document can be processed by an XML processor.

[First Hit](#) [Fwd Refs](#)

Generate Collection

Print

L3: Entry 1 of 9

File: USPT

Apr 20, 2004

DOCUMENT-IDENTIFIER: US 6725231 B2

TITLE: DICOM XML DTD/schema generator

Detailed Description Text (9):

In a preferred embodiment, the raw XML documents are processed via an XSLT (Extensible Stylesheet Language Transformation) engine 160. The additional advantage of segregating the XML-conversion from the XML-formatting is that existing XML-transformation tools and techniques can be used to effect the desired output XML format structure. In this preferred use of XSLT, the desired output XML format is specified using XSLT stylesheets 150. These stylesheets 150 are defined based on the format of the DICOM-XML DTDs and Schemas that is intended to be used for the development of one or more application programs. If a DICOM-XML standard is adopted for DICOM processing applications, then the use of stylesheets 150 that are compatible with this standard will allow the DICOM-XML DTDs and Schemas that are produced by the conversion system 100 to be used in the development of each application that is compatible with the standard. If a variety of DICOM-XML formats are defined, a different set of stylesheets 150 can be provided for each format, and thereby allowing the use of the same builder 130, regardless of the particular output format.

Detailed Description Text (24):

Referring again to FIG. 2, after conversion of each table in the DICOM specification 110 to a corresponding XML document 140 (140a-c), the XSLT engine 160, which may be any of a variety of available XSLT engines, provides the desired XML DTD and Schema output formats. As is known in the art, XSLT is a language that facilitates the transformation of an XML document into another XML document, using template matching. The stylesheets 150 contain template pairs. The original XML document is searched for a pattern that matches the first template in the pair. When the search results in a match, the information at the match location in the original XML document is converted to the form of the second template in the pair, and provided to the output XML document. In the subject invention, the first template is configured to match the form of the information in the documents 140a-c, and the second template is configured to correspond to whatever format is desired for use in a particular application, or, as noted above, to an agreed-upon standard format, for compatibility among a variety of applications.

Current US Original Classification (1):707/102Current US Cross Reference Classification (1):707/1

[First Hit](#) [Fwd Refs](#)

Generate Collection

Print

L3: Entry 3 of 9

File: USPT

Mar 16, 2004

DOCUMENT-IDENTIFIER: US 6708186 B1

TITLE: Aggregating and manipulating dictionary metadata in a database system

Detailed Description Text (7):

XML (Extensible Markup Language) is a standard language for defining application / industry-specific tagged dialects for description of the definer's unique data. XML is particularly useful for this purpose because it separates the description of the dialect from the content of the data for which the dialect is to be used. The separation of description and content permits easy parsing and transformation of XML documents using industry standard tools. For more information about XML, see <http://www.w3.org/XML>. In other embodiments, other intermediate representations may be employed. XML documents may be transformed by means of style sheets written in XSL. In its most general form, XSL is a way to transform an XML document into something else based on a set of rules embodied in the stylesheet. The output is another XML document, or HTML, or SQL text, or whatever else may be specified by the stylesheet. Recently, it has been recognized that the data transformation aspects of XSL are so powerful that they have been separated from the presentation rules and renamed XSL-T. For details, see <http://www.w3.org/Style/XSL>.

Detailed Description Text (10):

The remaining components of aggregation metadata 109 permit refinement of query 119 and specification of the transformations produced by XML transformer 127. Filter specifications 113 are specifications for restricting the set of objects for which the query 119 retrieves the metadata. For example, aggregate view 111 might be for the class of TABLE objects; a filter specification 113 can be used to limit the information retrieved by query 119 to the metadata for the set of TABLE objects whose names include the word PAYROLL. When a filter is specified in API 120, query generator 123 adds a WHERE clause to query 119 which specifies that the returned result will be limited to a subset of the rows in aggregation view 111 that have the property specified in the filter. Transform specifications 115 are specifications that describe transformations made by XML transformer 127 on the XML produced from the set 121 of aggregation objects; one such transformation is the transformation from XML to DDL.

Current US Original Classification (1):707/102Current US Cross Reference Classification (1):707/6

[First Hit](#) [Fwd Refs](#)

Generate Collection

Print

L3: Entry 7 of 9

File: USPT

May 6, 2003

DOCUMENT-IDENTIFIER: US 6560633 B1

TITLE: Method for creating network services by transforming an XML runtime model in response to an iterative input process

Detailed Description Text (123):XML Transformation FeatureDetailed Description Text (124):

One or more embodiments of the invention provide for a versatile service that performs a transformation of an XML tagged data object, using another tagged data object containing a block of XSL instructions. The service outputs a transformed tagged data object representing the transformed data. The service will wrap a standard XML and XSL processing engine such as the one from IBM called LotusXSL.

Detailed Description Text (127):

At regeneration, such a feature service dynamically builds the XSL block which is executed at runtime when the XML transformation service is called.

Detailed Description Text (131):

For illustrative purposes, the following description assumes that part of the "cellcatalog" template (the template utilized in accordance with one or more embodiments of the invention) has already been created. For example, it is assumed that the template already has a number of service calls and UI control features that present an XML cell phone catalog in a table on an input page. The base functionality provided in the template includes the ability to open a cell phone XML data source, and transform the data source into a catalog using an XML transformation service. In addition, the functionality includes the ability to extract the cell phone brand name list from the XML data, and presenting the information in drop down list 710 so that the user can select a brand, and filter the cell phone catalog by brand. Thus, the existing template provides for the display of a static HTML page (with the ability to select and display a specific brand of cellular phones) without a product info column containing Details buttons 702.

Current US Cross Reference Classification (1):707/2

Freeform Search

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term:

(707/\$.ccls.) and (transformation near xml)

Display:

50

Documents in Display Format:

FRO

Starting with Number

1

Generate: ☐ Hit List ☒ Hit Count ☐ Side by Side ☐ Image

Search

Clear

Interrupt

Search History

DATE: Thursday, June 10, 2004 [Printable Copy](#) [Create Case](#)**Set Name Query**

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=YES; OP=OR

| | | | |
|-----------|--|------|-----------|
| <u>L3</u> | (707/\$.ccls.) and (transformation near xml) | 9 | <u>L3</u> |
| <u>L2</u> | L1 and (transformation near xml) | 10 | <u>L2</u> |
| <u>L1</u> | 715/\$.ccls. | 5116 | <u>L1</u> |

END OF SEARCH HISTORY